CLAIMS

What is claimed is:

Sub a¹

~~1. A system for rendering graphics primitives using a shortened display list for ensuring a shortened processing time while maintaining the quality of the display information contained in the display list, the system comprising:~~

~~a system bus for communicating data and instructions;~~

~~a host processor coupled to the system bus for processing a display list defining a graphics primitive;~~

~~a system memory coupled to the system bus for storing the display list;~~

~~a graphics subsystem coupled to the system bus for processing display parameter values contained in the display list, wherein the display list includes a field load instruction for effectively shortening display list processing; and~~

~~a display unit coupled to the graphics processor for displaying the graphics primitives comprising the display list.~~

~~2. The system of claim 1 further including a frame buffer coupled to the graphics processor for storing the display list.~~

B

~~3. The system of claim 1, wherein the graphics subsystem includes a plurality of storage means for storing the parameter values representative of the graphics primitives in the display list.~~

~~4. The system of claim 3, wherein the graphics subsystem further includes a address generating means for generating address offsets bits responsive to the display parameter values contained in the display list.~~

~~5. The system of claim 4, wherein the graphics subsystem further includes an instruction storing means for storing a plurality of instruction data bits responsive to each of the plurality of display list instructions, said instruction data bits shortened to allow the display list to be processed within a shortened processing cycle in the graphics subsystem.~~

B

~~6. The system of claim 5, wherein the graphics subsystem further includes an instruction sequencing means coupled to the instruction bits storage means for sequencing bits of the display list into a plurality of register files.~~

~~7. The system of claim 6, wherein the graphics subsystem further includes an instruction fetch means for randomly fetching a next display list parameter value for a display primitive to be processed in the graphics subsystem.~~

~~8. The system of claim 7, wherein the graphics subsystem further includes an address counting means for sequentially counting the~~

~~address storage locations for the display list parameter values of the graphics primitives stored in the plurality of register files.~~

6

~~9. The system of claim 8, wherein the graphics subsystem further includes an address partition storage means for storing addresses responsive to load instructions representing the shortened display list, and wherein portions of the instruction storing means index the address partition storage means.~~

Sub R

~~10. A graphics system for processing parameter values of graphics primitives in a display list, wherein the display list is shortened to enable fast processing time while maintaining the quality of information contained in the display list, the graphics system comprising:~~

~~a plurality of register files for storing a plurality of parameter values representing graphics primitives defined in the display list; and~~

~~a graphics processor coupled to the plurality of register files, wherein the graphics processor processes the shortened display list while maintaining the display quality of primitives displayed in a display unit.~~

~~11. The graphics processor of claim 10 including an instruction fetch logic unit for fetching the next parameter values responsive to a graphics primitive to be displayed.~~

~~12. The graphics processor of claim 11 further including a load instruction unit for storing load instructions representative of a shortened display list instruction, said load instruction comprising a plurality of data bits each of said plurality of data bits representing specific load functions to be performed by the load instruction.~~

C

~~13. The graphics processor of claim 12, wherein the load instruction unit includes an opcode storage unit for storing opcode information responsive to each of the load instructions for determining the type of function to be performed by the graphics primitive to be rendered.~~

Sub B3

~~14. The graphics processor of claim 13, wherein the load instruction unit further includes a write enable storage portion for storing write enable data for determining whether to load one of the plurality of registers in the register.~~

~~15. The graphics processor of claim 14, wherein the load instruction unit further includes an instruction partition portion for storing partition data for referencing the partition table to load parameter values to the referenced register.~~

C

Sub B3

~~16. The graphics processor of claim 10 further comprising a data shifter coupled to the load instruction unit for sequentially~~

~~shifting data bits corresponding to a load instruction in order to write the load instructions to the register files.~~

~~17. The graphics processor of claim 16 further comprising an address counter coupled to the register files for sequentially counting the address offsets of the register files locations as the load instruction data is loaded into the register files.~~

~~18. The graphics processor of claim 17 further comprising a partition table coupled to the load instruction unit for storing the address offset bits corresponding to random register locations in the register files for the display parameter values in the display list.~~

Sub B34

~~19. The graphics processor of claim 18 further comprising a write enable signal coupled to the address bit shifter, said write enable signal asserted high to allow the graphics processor to write the load instructions to the register files, wherein the write enable signal enables the graphics processor to randomly load register locations in the register file.~~

~~20. The graphics processor of claim 19 further comprising a request next parameter value signal coupled to the fetch logic unit, said request next parameter signal asserted high to allow the next parameter value in the display list to be fetched by the graphics processor.~~

Sub B5

~~21. The graphics processor of claim 20 wherein the partition look-up table comprises 64 entries of address offsets to the register file.~~

~~22. The graphics processor of claim 21 wherein the 64 entries of the partition table are evenly distributed to corresponding register locations in the register file.~~

~~23. The graphics processor of claim 22 wherein each of the 64 entries of the partition table is 6 binary wide.~~

~~24. The graphics processor of claim 23 wherein the register file comprises 1024 entries of addresses.~~

Sub B3

Claims 25-30 305/566 KW 8/13/96

~~25 A method of encoding and decoding a shortened display list load instruction comprising the steps of:~~

~~encoding a field load instruction wherein the field load instruction comprises an opcode instruction, a write enable field and a partition index field;~~

~~loading the field load instruction to a register file; and~~

~~executing the field load instruction in a shortened processing time.~~

~~26 The method of claim 25 wherein the field load instruction execution step comprises the step~~ B of enabling the write enable

~~field to allow the load instruction to be randomly load to the register file.~~

~~27. The method of claim 25 wherein the loading step comprises the step of loading a shift register with write enable data from the field load instruction.~~

~~28. The method of claim 27 wherein the instruction loading step further comprises the step of indexing a partition look-up table with write enable data from the write enable field.~~

B

~~29. The method of claim 27 wherein the instruction loading step further includes loading an address counter to sequentially count the number of load instructions in the display list.~~

~~30. The method of claim 29 wherein the write enable field is disabled to skip the loading of a register in the register file.~~

Add C1